Efficient classical simulation of the Deutsch-Jozsa algorithm

Niklas Johansson,* Jan-Åke Larsson†
Institutionen för Systemteknik, Linköpings Universitet, 581 83 Linköping, SWEDEN

In 1985, David Deutsch challenged the Church-Turing thesis by stating that his quantum model of computation *"could, in principle, be built and would have many remarkable properties not reproducible by any Turing machine"*. While this is thought to be true in general, there is usually no way of knowing that the corresponding classical algorithms are the best possible solutions. Here we provide an efficient classical simulation of the Deutsch-Jozsa algorithm, which was one of the first examples of quantum computational speed-up. Our conclusion is that the Deutsch-Jozsa quantum algorithm owes its speed-up to resources that are not necessarily quantum-mechanical, and when compared with the classical simulation offers no speed-up at all.

The Deutsch-Jozsa algorithm [1, 2] was one of the first indications of quantum computational speed-up [3]. The algorithm has, since then, been used extensively for illustrating experimental realizations of a quantum computer and the quantum computational speed-up. The size of this speed-up crucially depends on the complexity of the corresponding most efficient classical algorithm. Here we present an algorithm within an extension of Spekkens' "toy theory" ([4], here: toy model), that deterministically solves the Deutsch-Jozsa problem with only one oracle query. Relative to the oracle, the algorithm is efficiently simulatable on a classical Turing-machine, making it clear that there is no quantum speed-up in this case. In the language of complexity classes, the algorithm no longer gives any evidence of an oracle separation between EQP (Exact or Error-free Quantum Polynomial time solvable problems [5]) and P (Polynomial time solvable problems).

Suppose that you are given a Boolean function $f(x) : \{0,1\}^n \mapsto \{0,1\}$ with the promise that it is either constant or balanced. The function is constant if it gives the same output (one or zero) for all possible inputs, and it is balanced if it gives the output zero for half of the possible inputs, and one for the other half. Your task is now to distinguish between these two cases [2, 6]. Given such a function, a classical Turing-machine can solve this problem by checking the output for $2^{n-1} + 1$ values of the input; if all are the same, the function is constant, and otherwise balanced. A stochastic algorithm with k randomized function queries gives a bounded error probability [2] less than 2^{1-k}.

An oracle for a quantum computer, a quantum oracle, is a unitary transformation that implements the function. Given such an oracle, a quantum computer can solve the problem with a single query by using the Deutsch-Jozsa algorithm [2, 6], in the ideal case with zero error probability. Figure 1 shows a quantum-circuit representation of the algorithm: prepare an n-qubit input-register, a 1-qubit target, and put them through Hadamard transformations to produce a full superposition over all computational basis states. Perform the oracle transformation U_f, corresponding to addition modulo 2 of the function value to the target qubit, and finally apply Hadamard transformations to restore the input-register. This procedure will leave the input-register unchanged only when the function is constant, otherwise it will have changed [2, 6]; a measurement of the input-register will reveal this.

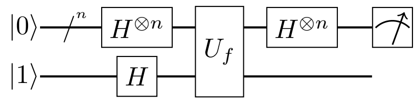

Figure 1: Circuit performing the Deutsch-Jozsa algorithm. This circuit uses an n-qubit input-register prepared in the state $|0\rangle^{\otimes n}$, and a target prepared in $|1\rangle$. It proceeds to apply Hadamard transformations to each qubit. The function f is embedded in an oracle U_f, and this is followed by another Hadamard transformation on each qubit. The measurement at the end will test positive for $|0\rangle^{\otimes n}$ if f was constant, and negative if f was balanced.

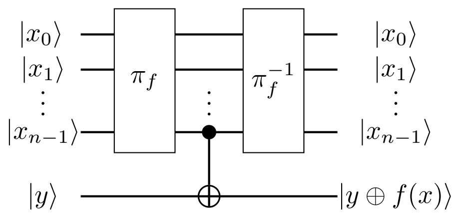

Figure 2: Oracle construction. The n-qubit gates at the beginning and end are unitary permutations of computational basis states ($\pi_f^{-1} = \pi_f^\dagger$). At the center is a CNOT gate from the most significant qubit $|x_n\rangle$ to the target $|y\rangle$.

For $n = 1$ and $n = 2$ qubits in the input-register, the Deutsch-Jozsa oracle is known to have an implementation that does not rely on quantum resources [7]. These two cases can be implemented [8] in the framework of Spekkens' toy model [4] that captures many, but not all,

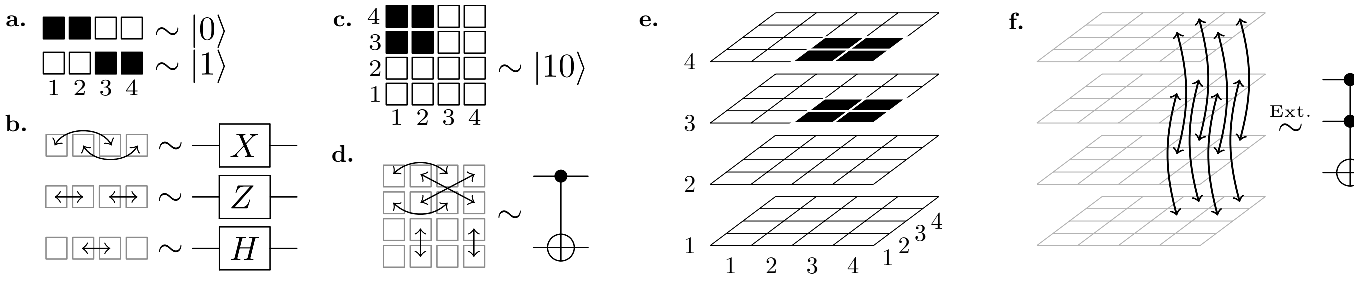

Figure 3: States and transformations. **a.** Epistemic states of an elementary system encoding logical 1 and 0. **b.** Permutations corresponding to Pauli-X, Pauli-Z, and Hadamard transformations. **c.** Composite system of two elementary systems in an epistemic state corresponding to $|01\rangle$. **d.** Permutation corresponding to a CNOT gate with the vertical system as control and the horizontal system as target. **e.** A tripartite system in an epistemic state corresponding to $|101\rangle$, with the elementary system along the width and height as the most and least significant systems. **f.** Transformation corresponding to the permutation $|110\rangle \longleftrightarrow |111\rangle$, note that this is not a valid permutation in Spekkens' toy model.

properties of quantum mechanics. The toy model encodes the state of each elementary system (or toy bit) into a one-out-of-four *ontic state*. The toy model follows the *knowledge balance principle*; stating that at every point in time at most one bit of information per toy bit can be known or predicted by an observer. Therefore, a measurement can only yield knowledge about which *epistemic state* the system is in, corresponding to a uniform probability distribution over the relevant part of the ontic state space. The epistemic state is closely related to the quantum state. For example, the epistemic state associated with $|0\rangle$ is a uniform distribution over the ontic states 1 or 2, written $1 \vee 2$. Each even partition of the state space corresponds to a basis of the quantum state space as indicated in Figure 3a. Unitary transformations are modeled by ontic state permutations, as indicated in Figure 3b. Systems compose under the Cartesian product (see Figure 3c).

This construction even captures some properties of entanglement, enabling protocols like super-dense coding and quantum-like teleportation, but cannot give all consequences of entanglement, most importantly, it cannot give a Bell inequality violation. However, the toy model's epistemic states and transformations are closely related to stabilizer states and Clifford group operations [9], including the CNOT gate as shown in Figure 3d. Indeed, a non-trivial subset of quantum mechanics can be efficiently simulated on a classical Turing machine via the Gottesman-Knill theorem [10]. Furthermore, classical Turing machine simulation of Spekkens' toy model (of at most a polynomial size in the number of subsequent operations) is efficient [4]; all operations are time efficient, and the four ontic states of an elementary system is stored in two classical bits; an n-toy-bit system therefore requires $2n$ bits of storage.

However, for $n \geq 3$, the Deutsch-Jozsa oracle needs the Toffoli gate, and since the Toffoli gate is not efficiently simulatable using the stabilizer formalism [10] nor present in the toy model [4], it has so far been believed that the Deutsch-Jozsa algorithm does not have an efficient equivalent in the toy model. Here, we need to point out that our task is not to create Toffoli gate equivalents in the toy model, or even simulate the quantum-mechanical system as such. It suffices to give a working efficient toy model equivalent of the Deutsch-Jozsa oracle. We therefore choose not to represent Toffoli gates exactly, but design the ontic state permutation so that it swaps the epistemic states associated with $|110\rangle$ and $|111\rangle$, but does not permute the ontic states within these computational-basis epistemic states, see Figure 3f. This type of transformations are, in general, not valid transformations in Spekkens' toy model [4], but as already stated, our aim is not to simulate quantum mechanics, it is to devise an equivalent of the Deutsch-Jozsa algorithm. The extended model is still efficiently simulatable on a classical Turing machine.

By representing a general computational-state permutation

$$\pi = \sum_x |\pi(x)\rangle\langle x| \tag{1}$$

in the same manner as the Toffoli we obtain an extended toy model oracle, proving the existence of such an oracle in this extended model. The toy model permutation reminiscent of Hadamard gates [9] can now be used, and the Deutsch-Jozsa algorithm works as in the quantum case. Figure 2 shows an oracle construction for balanced functions, where π_f is an arbitrary permutation of the computational basis states which can be constructed from CNOT and Toffoli gates [11]. At the center of the oracle a CNOT gate is applied between the most significant qubit in the input-register and the target. With π_f as the identity permutation, the oracle performs the balanced function f' that is 1 for all inputs with the most significant bit set. Any other balanced function can now be generated by choosing a different π_f, giving the function output $f(x) = f'(\pi_f(x))$. For a function that is constant zero, the CNOT gate is omitted, so that the target value is unchanged independently of the input-register. The alternative constant one function can be created by replacing the CNOT with a Pauli-X gate acting on the target, inverting the target independently of the input-register.

This device will give the expected outcomes for classical function queries, i.e., invocations that reveal a single function value. The desired input number x should be inserted in the input-register as the quantum state $|x\rangle$ in the computational basis, and the target should be in a computational basis state (say $|0\rangle$). Applying the oracle and measuring the target will reveal the output of f: if the target was flipped, then the function value is one for that input x.

The following is a simple description of the resulting complete toy-model oracle in terms of a single transformation of the ontic state space. For any balanced function the oracle will have the following effect:

1. The input-register ontic state belongs to some computational basis epistemic state associated with $|x\rangle$. If $f(x) = 1$, perform a Pauli-X transformation on the target.

2. If the target system is in ontic state 2 or 4, perform a Pauli-Z on the most significant input-register toy bit.

The constant-function oracles do not have the CNOT, and will keep the input-register ontic state stationary for all possible inputs, because the permutation π_f is immediately followed by its inverse. The target ontic state may change, depending on the actual constant function.

In our extended toy model algorithm we now prepare the input-register in the epistemic state $(1\vee 2)^n$ that corresponds to $|0\rangle^n$, and the target in $3\vee 4$ that corresponds to $|1\rangle$, so that the whole system's epistemic state is

$$\big(\underbrace{1\vee 2, 1\vee 2, \dots, 1\vee 2}_{n\text{ elementary systems}}, 3\vee 4\big)\ . \tag{2}$$

The Hadamard transformation gives us

$$\big(\underbrace{1\vee 3, 1\vee 3, \dots, 1\vee 3}_{n\text{ elementary systems}}, 2\vee 4\big). \tag{3}$$

We see that the target system is guaranteed to be in an even ontic state.

Applying the oracle for constant functions, the input-register will stay unchanged, since the CNOT is not present, alternatively replaced with a Pauli-X gate on the target toy bit. In this case, a second Hadamard transformation on each toy bit will return the input-register to the initial epistemic state, while there may be a change in the ontic state of the target toy bit.

Applying the oracle for balanced functions the CNOT in the oracle will induce a Pauli-Z permutation of the most significant system in the input-register.

$$\big(\underbrace{2\vee 4, 1\vee 3, \dots, 1\vee 3}_{n\text{ elementary systems}}, 2\vee 4\big) \tag{4}$$

A second Hadamard transformation on the input-register will then produce the epistemic state

$$\big(\underbrace{3\vee 4, 1\vee 2, \dots, 1\vee 2}_{n\text{ elementary systems}}, 2\vee 4\big) \tag{5}$$

Measurement of the input-register will now reveal whether the function was constant (the epistemic state completely overlaps with the initial epistemic state) or balanced (the epistemic state is disjoint with the initial epistemic state).

This system deterministically solves the problem using only one oracle query. It can be efficiently simulated by a classical Turing machine with an increase in memory (comparing qubits/toy bits with classical bits) by only a constant factor of two. The spatial and temporal complexity of this simulation are therefore identical to the complexity of the quantum algorithm. We do recognize that this solution differs from the quantum solution in the sense that the input epistemic state is mapped to the same output epistemic state for all balanced functions, which is not necessarily the case with the quantum solution. Work has been done on a gate representation [8] that mimics this behaviour more closely, but this does not reach zero error probability, essentially because the quantum Toffoli gate does not have an implementation in the toy model.

In conclusion, we have devised a toy model equivalent of the Deutsch-Jozsa quantum algorithm, which is efficiently simulatable on a classical Turing machine. In the quantum algorithm, you are given an oracle that implements a function f, your task is to distinguish balanced from constant functions. In the presented classical simulation of the algorithm, you are given an oracle that implements the the function in the same way, but within the framework of the extended toy model, and your task is still to distinguish balanced from constant functions. By this method we obtain equal temporal and spatial complexity as for the quantum algorithm. It is possible that the same technique can be used for other quantum oracle algorithms, but these have been conjectured to use genuinely quantum properties such as the continuum of quantum states, or contextuality [12, 13], which both are missing from the toy model [4]. Therefore it remains to be seen what efficiency can be achieved for these other algorithms. However, it is clear that the Deutsch-Jozsa algorithm does not need genuinely quantum properties to work, and that the Deutsch-Jozsa algorithm shows no speed-up at all compared with this new toy model algorithm or its classical simulation.

Acknowledgments.— The project has been supported by the Foundational Questions Institute (FQXi) through the Silicon Valley Community Foundation.

Supplementary Information.— Available in the form of a classical simulation of the toy model algorithm, in Python, supplied with the online version of the paper.

* Electronic address: niklas.johansson@liu.se
† Electronic address: jan-ake.larsson@liu.se

[1] D. Deutsch, "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer", Proc. Roy. Soc. London A **400**, 97–117 (1985) `10.1098/rspa.1985.0070`.

[2] D. Deutsch and R. Jozsa, “Rapid Solution of Problems by Quantum Computation”, Proc. Roy. Soc. Lond. A **439**, 553–558 (1992) `10.1098/rspa.1992.0167`.

[3] D. R. Simon, “On the power of quantum computation”, in Proceedings of the 35th Annual Symposium on Foundations of Computer Science (1994), pp. 116–123, `10.1109/SFCS.1994.365701`.

[4] R. W. Spekkens, “Evidence for the epistemic view of quantum states: A toy theory”, Phys. Rev. A **75**, 032110 (2007) `10.1103/PhysRevA.75.032110`.

[5] E. Bernstein and U. Vazirani, “Quantum Complexity Theory”, SIAM J. Comput. **26**, 1411–1473 (1997) `10.1137/S0097539796300921`.

[6] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, “Quantum algorithms revisited”, Proc. Roy. Soc. London A **454**, 339–354 (1998) `10.1098/rspa.1998.0164`.

[7] D. Collins, K. W. Kim, and W. C. Holton, “Deutsch-Jozsa algorithm as a test of quantum computation”, Phys. Rev. A **58**, R1633–R1636 (1998) `10.1103/PhysRevA.58.R1633`.

[8] N. Johansson, *Efficient Simulation of Deutsch-Jozsa Algorithm* (Linköping University, The Institute of Technology, 2015), ISRN: LiTH-IFM-A-EX--15/2992--SE.

[9] M. F. Pusey, “Stabilizer Notation for Spekkens’ Toy Theory”, Found. Phys. **42**, 688–708 (2012) `10.1007/s10701-012-9639-7`.

[10] D. Gottesman, “The Heisenberg Representation of Quantum Computers”, (1998), arXiv:`quant-ph/9807006`.

[11] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary Ed. (Cambridge University Press, New York, USA, 2011), ISBN: 1107002176.

[12] J.-Å. Larsson, “A contextual extension of Spekkens’ toy model”, in AIP Conference Proceedings, Vol. 1424 (2012), pp. 211–220, `10.1063/1.3688973`.

[13] M. Howard, J. Wallman, V. Veitch, and J. Emerson, “Contextuality supplies the magic for quantum computation”, Nature **510**, 351–355 (2014) `10.1038/nature13460`.